# SHAP 値や重要度を用いたモデル解釈性: 包除積分ネットワークと XGBoost の比較

板橋将之[1]　本田あおい[2]　大北剛[2]

**概要**：包除積分ネットワークの多項成分の及ぼす性能を測定するため，SHAP 値や重要度などの分類モデルを解釈するツールを用いて， XGBoost との比較を行った．

**キーワード**：ニューラルネットワーク，機械学習

# Model Interpretability Using SHAP values and Importance:Comparing Inclusive Integration Networks and XGBoost

MASAYUKI ITABASHI[†1]　AOI HONDA[†2]　TSUYOSHI OKITA[†2]

***Abstract***：***To measure the performance of the polynomial components of the inclusion integral network, the SHAP value A comparison with XGBoost was performed using a tool that interprets classification models such as degrees.***

***Keywords***:Neural Network, Machine learning, explainable AI

## 1. 序論

機械学習の解釈性はベイズ機械学習では概ね問題ないと考えられているが，頻度論の機械学習では大きな問題となっている．これは説明可能性 AI という言葉で取り上げられることが多い．学習の結果について，何を考慮され結果が導かれたのかという根拠があまり示されないことによる．これに対して，古典的なシャープレイ値や重要度をはじめとする機械学習に対する因子の情報を明示的に表明することにより，これを解釈性のボトムラインととる捉えられるやり方がある．すべての特徴の重要度を羅列するやり方は非常にプリミティブなため，現実的に用いられる万や 10 万という特徴の存在する横長データなどでは無力となる．現在，人工知能の分野における説明可能性 AI は，局所的な説明可能性と大域的可能性を用いることにより，病気の診断や，なぜ画像をいかにうまく解釈するかが研究されている．

包除ネットワークは，隠れ層 1 層の機械学習アルゴリズムであるが，この解釈性の問題にプリミティブな形でどう対処するかを本論文では見ていくことを目的とする．目標は古典的なシャープレイ値や重要度を用いる形を 2 つのデータ集合を用いて実験を行なう．

## 2. 包除積分ネットワーク

訓練集合を$S = (x, y)_{i=1}^{n}$とし，x と y いずれも実数集合を値域とする．訓練標本は S から独立同一に抽出されたと仮定する．損失関数は回帰問題に対しては最小二乗損失，分類問題に対しては，交差エントロピー損失を考える．

ネットワークの構造は通常の前進型ニューラルネットワーク(feedforward neural network)に類似するが，入力因子相互の 1 次のペアワイズな依存関係を考慮する部分が異なる．このアーキテクチャは入力因子相互の 2 次のペアワイズな依存関係を考慮するアーキテクチャと類似する点も多いが，(1)$X^2$の項が存在しない点，(2)多項式$x_1, x_2, .. x_n$などの異なる因子であればそれらの相互作用が考慮されている点(つまり，2 次以上の 3 次や 4 次も考慮し

1 九州工業大学　大学院　学際情報工学専攻
Kyushu Institute of Technology
2 九州工業大学
Kyushu Institute of Technology

ている点)，という大きな違いが存在する．このような特殊なアーキテクチャを考慮する背景には包除積分がある．

隠れ層 1 層を含むニューラルネットワークを構築する．入力層と隠れ層の結合においては，通常のニューラルネットワークで用いられる結合に加えて，入力因子相互の互いに素な N 項ワイズな依存関係を考慮する．さらに，隠れ層と出力層の結合においては， 通常のニューラルネットワークで用いられる結合の出力以外に，入力因子相互の互いに素な N 項ワイズな依存関係の出力を考慮する． 通常のニューラルネットワークで行なわれるように，それぞれの因子において前層の出力を単純に重みつけ総和をするアーキテクチャとする．

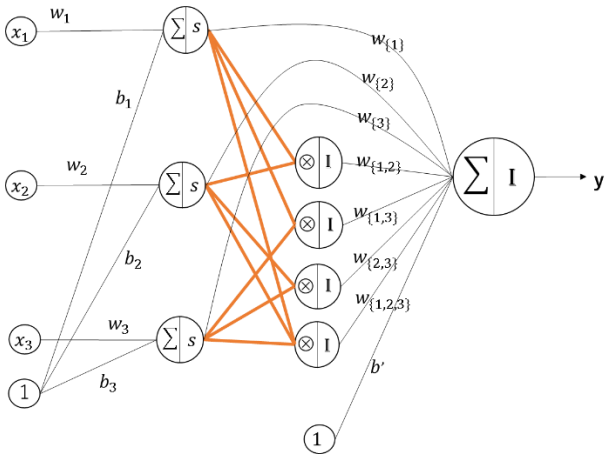


図 1　包除積分ネットワーク

s:シグモイド関数
b':バイアス
w:重み
I:恒等関数

包除積分ネットワークとは包除積分をニューラルネットワークで表現したものである．包除積分は「$\Omega=\{1,2,\ldots,n\}$は有限集合とする．$|A|$は有限集合 A の要素の項数を，$P(\Omega)$は$\Omega$の部分集合全体からなる集合，つまり $\Omega$ のべき集合を表す．

$\Omega$ 上の集合関数 $\mu: P(\Omega)\to[0,\infty]$は
(i)$\mu(\emptyset)=0$,
(ii)$A,B\in P(\Omega)$で $A\subset B$ ならば $\mu(A)\leqq\mu(B)$
を満たすとき，$\Omega$ 上のファジィ測度という．
被積分関数 f は $\Omega$ 上に定義された$[0,K]$に値をとる非負有界値関数である．今，$\Omega$ は n 個の要素から成る有限集合なので，$\Omega$ 上の関数は n 次元の数ベクトルで表せる．これを$f=(x_1, x_2,.. x_n)\in[0,K]^n$と書くことにする．
$\Omega$ 上の非負有界関数$f=(x_1, x_2,.. x_n)\in[0,K]^n$の $\mu$ と$\otimes$による包除積分は次で定義される．

$$\otimes\int f\,d\mu:=\sum_{A\in P(\Omega)\setminus\{\emptyset\}} M^{\otimes}(f|A)\mu(A),$$

ただし，

$$M^{\otimes}(f|A):=\sum_{B\in P(\Omega),B\supset A}(-1)^{|B\setminus A|}\otimes_{i\in B}x_i,$$

メビウスの反転公式を用いると包除積分は次のように別表現できる．

$$\otimes\int f\,d\mu=\sum_{A\in P(\Omega)\setminus\{\emptyset\}}(\otimes_{i\in A}x_i)m^{\mu}(A),$$

ただし$m^{\mu}$は $\mu$ のメビウス変換：

$$m^{\mu}(A):=\sum_{B\subset A}(-1)^{|B-A|}\mu(B).$$

」[1]と定義される．今回は K=1 として⊗は掛け算を用いる．例として$\Omega=\{1,2,3\}$の場合で包除積分の式を書き下す．簡単のため$a_A:=m^{\mu}(A)$, と書くことにすると:

$$\begin{aligned}(AP)\int f\,da &= a_{\{1\}}x_1+a_{\{2\}}x_2+a_{\{3\}}x_3\\ &\quad+a_{\{1,2\}}x_1x_2+a_{\{1,3\}}x_1x_3+a_{\{2,3\}}x_2x_3\\ &\quad+a_{\{1,2,3\}}x_1x_2x_3+a_{\emptyset} \qquad \cdots(1)\end{aligned}$$

となる．これは重回帰モデルに第 4 項から第 7 項の交互作用項を加えたものとみなすことができる
図 1 の包除積分ネットワークではシグモイド関数を活性化関数としているユニットの出力を$x_i'$と置きなおすことで，出力$y$ の値を，

$$\begin{aligned}y &= w_{\{1\}}x_1'+w_{\{2\}}x_2'+w_{\{3\}}x_3'+w_{\{1,2\}}x_1'x_2'\\ &\quad+w_{\{1,3\}}x_1'x_3'+w_{\{2,3\}}x_2'x_3'\\ &\quad+w_{\{1,2,3\}}x_1'x_2'x_3'+b' \qquad \cdots(2)\end{aligned}$$

とすることができる．この出力の式が（1）の包除積分の式を表している.今回はこの包除積分ネットワークの(2)の重みの値からシャープレイ値という変数ごとの貢献度を求めることで解釈していく．

## 3. XGBoost

決定木とは木を逆さにしたようなデータ構造を用いて分類と回帰行う機械学習の手法で，決定木を複数使って行われる学習（アンサンブル学習）の中の一つに XGBoost[2]というものがある．アンサンブル学習には，
Bagging:様々な弱学習機を独立に作り，それらの予測結果を統合する手法
Boosting:様々な学習機を逐次的に作り，それらの予測結果を統合する手法
という 2 種類の方法があり，Bagging には Random Forest などの手法があり，後者には AdaBoost や Gradient Boosting などの手法が存在し，XGBoost は後者に属している．Boosting では 2 値分類の場合以下のような流れで強学習器を作っていく．
訓練データ$(x_i, y_i)$（ただし$i = 1,\ldots,n$）をもとに，二値分類器$f(x)$を作る．$y_i$は 1 か-1 として，各分類器$f$ の出力も 1 か-1 とする．

1．　弱学習機を作る．

訓練誤差$E_1=\frac{1}{n}\sum_{i=1}^{n}[y_i\neq f_1(x_i)]$が最小となるように$f_1(x_i)$を作る．ただし，$[y_i\neq f_1(x_i)]$は$y_i\neq f_1(x_i)$の時 1，それ以外 0 をとるものとする．

2．　$f_1(x)$の結果を考慮して，次の弱学習機$f_2(x)$を作る

訓練誤差$E_2 = \frac{1}{n} w_i^{(2)} \sum_{i=1}^{n} [y_i \neq f_2(x_i)]$が最小となるように$f_2(x_i)$を作る．$w_i^{(2)}$は$f_2$作成時の各サンプルの重みを表し，$f_1(x_i)$が誤分類したサンプルの重みが大きくなるように計算される．

3． 順々に$f_{t-1}(x)$の結果を考慮して，次の弱学習機$f_t(x)$を作る

4． $f_k(x)$まで作ったら，$f_1(x)$から$f_k(x)$までの出力をまとめて強学習機$f\ (x)$を作る

$f\ (x) = sign\{\sum_{t=1}^{k} \alpha_t f_t(x)\}$として弱学習器をまとめる．$\alpha_t$は t 番目の学習器にかける重みで，sign は符号関数（正の入力には 1，負は-1 を返す）．この時$\alpha_t$は

$\alpha_t = \frac{1}{2} \ln(\frac{1-E_t}{E_t})$という値を使う．（$E_t$は学習器$f_t$の誤差）

t 回目の$w_i^{(t)}$は$w_i^{(t)}$=$C_t w_i^{(t-1)} e^{-y_i \alpha_t f_t(x_i)}$とする．また

$C_t = \frac{1}{\sum_{i=1}^{n} w_i^{(t-1)} e^{-y_i \alpha_t f_t(x_i)}}$は正規化定数である．$f_t$が$x_i$を誤分類したときには，そのサンプルの重みが大きくなるように更新（$C_t e$倍）し，正しく分類できた時は，そのサンプルの重みが小さくなるように更新（$C_t e^{-1}$倍）する．(t-1)番目までの学習が終わったうえで$\sum_{i=1}^{N} e^{y_i\{\alpha_1 f_1(x_i) + \cdots + \alpha_t f_t(x_i)\}}$を最小にするような$\alpha_t$，$f_t$を求めると先ほどの

$w_i^{(t)}$=$C_t w_i^{(t-1)} e^{-y_i \alpha_t f_t(x_i)}$と$\alpha_t = \frac{1}{2} \ln(\frac{1-E_t}{E_t})$の 2 式が導出される．XGBoost では損失関数に微分可能な関数を使用するのに加え，木の構造を複雑になりすぎないように正則化項をつけて形が固定された木構造の最適解を近似的に求めていく．木構造は木の分岐をする前とした後の誤差を考慮していくことでどのような木構造にするのかを決めていく．さらに学習率に当たる Shrinkage で学習を遅らせる，Column Subsampling で入力データの一部を使うことで過学習を防止することもできる．

## 4. シャープレイ値,SHAP 値,重要度

### 4.1 シャープレイ値

ファジィ測度はメビウス逆変換:

$$\mu(A) = \sum_{B \subset A} m^{\mu}(B)$$

で計算でき，この値からどの項目を重視しているかわかる．しかし，ファジィ測度の値は単独での重視度や複数項目を合わせたものの重視度が得られるため各要因の重視度の比較が行いづらい．これは次式で示すシャープレイ値を計算することにより項目ごとの比較が可能になる:

$$\phi_i(\mu) = \sum_{A \subset \Omega \setminus \{i\}} \frac{|A|!(n-|A|-1)!}{n!} (\mu(A \cup \{i\}) - \mu(A)).$$

$\Phi(\mu) = (\phi_1(\mu), \phi_2(\mu), \ldots, \phi_n(\mu))$の$\phi_i$が項目 i の重視度で

$$\sum_{i=1}^{n} = \mu(\Omega)$$

が成り立つ．なお，単調性を満たしていない集合関数についてもシャープレイ値は計算可能である．

包除積分ネットワークでは重みの値からファジィ測度を求めることができ，この値からシャープレイ値を求める．

### 4.2 SHAP 値

SHAP[3]は深層学習モデルを説明する手法の一つとして知られており，機械学習で学習したモデルに対して可読表現を用いて説明する手法である．可読表現のモデルの条件として，機械学習で学習したモデルの出力を$f(x)$，説明用のモデルを線形モデルとし$g(x') = w_0 + \sum_i w_i x_i'$としたとき，

1. $f(x) = g(x')$
2. $x_i' = 0 \Rightarrow w_i = 0$
3. $f(x) = f_x(x')$ ただし，

$$f_x'(x') - f_x'(x' \setminus i) \geq f_x(x') - f_x(x' \setminus i) \Rightarrow w_i(f') \geq w_i(f)$$

の 3 条件を満たす説明モデルがただ一つ求められ，$d$=$x'$のサイズとすると，

$$w_i = \sum_{Z \subseteq x'} \frac{|z|!\,(d - |z| - 1)!}{d!} (f_x(z) - f_x(z \setminus i))$$

となり，これはシャープレイ値と等価となる．

SHAP 値の計算方法として重み付き最小二乗法を計算し最小となる $w$ を求める．式にすると，

$$\min_w \sum_{z \in Z} \pi(z)(f_x(z) - g(z))^2, \pi(z) = \frac{d-1}{{}_nC_{|z|}|z|(d-|z|)}$$

を計算することで求められる．また SHAP はモデルごとに特化して計算し，線形モデルの貢献度を計算する Linear SHAP，決定木を用いたモデルの貢献度を計算する Tree SHAP，深層学習モデルの貢献度を計算する Deep SHAP と機械学習でよく用いられるモデルの SHAP をそれぞれ求めることができる．

### 4.3 重要度

XGBoost では学習したツリーモデルから変数の重要度というものを得ることができる．XGBoost の重要度には 3 つの指標があり，

1. weight
   weight は作成したツリーモデルの中にその変数がいくつ分岐として使用されているかを示している．
2. gain
   各変数が平均的にどれだけ評価を改善させたのかを示している．
3. cover
   学習したツリーモデルの葉に分類された訓練データの 2 次勾配の合計値を示している．

今回行う実験では gain を用いて各変数ごとの重要度を示している．

## 5. クラス分類問題での不均衡データ問題

不均衡データとは，データセット内のクラスごとの割合

が大きく異なるようなデータセットのことである．不均衡データを用いて機械学習を行う際には適切な評価指標や学習を行わなければ真に求めたいクラス分類を行うことができない．この不均衡データ問題を改善するために既存のデータセットから新たにデータセットを抽出する手法として Under sampling と Over sampling というものがある．

### 5.1 Under Sampling

Under Sampling とはデータセットの多数派であるクラスのデータをいくつか取り出して多数派のクラスと少数派のクラスの比率を元のデータより均衡にする操作である．6 章の実験では Python のライブラリとして提供されている imblearn(1)の Cluster_Centroid と Random_under という Under Sampling の手法を用いている．Cluster Centroid では各多数派クラスのデータの重心を求め，データの特徴を崩さないようにデータを抽出している．Random_under では各多数派クラスからランダムにデータを抽出している．

### 5.2 Over Sampling

Over Sampling では Under Sampling の逆で少数派であるクラスのデータを使用して新たにデータを作成，追加し，データ数を増やし元のデータより均衡にする操作である．6 章の実験では Under Sampling でも使用した imblearn から提供されている SMOTE[4]という手法を用いている．SMOTE は少数派クラスのデータ同士を直線でつなぎ，その線分上からランダムにデータを作成することで少数派クラスのデータ数を増やしている．

## 6. 実験

実験では提案手法である包除積分ネットワークと既存の機械学習手法である XGBoost の二つでそれぞれ学習を行い，学習できたモデルからシャープレイ値と gain の値を求めていく．データは特徴の異なるデータセットを 2 つ用いてそれぞれのデータで実験を行った．

### 6.1 Car_evaluation データセットの準備

実験を行う際に使用するデータは，UCI Machine Learning Repository で機械学習のために公開されているデータセットの一つである Car_evaluation(2)というデータセットである．このデータは 6 つの変数からなり，

表 1　Car_evaluation の生データ

| 変数名 | 変数の値 |
|---|---|
| 総合評価 | unacc, acc, good, vgood |
| 買取価格 | vhigh, high, med, low |
| メンテナンス費 | vhigh, high, med, low |
| ドアの数 | 2, 3, 4, 5 |
| 乗車定員 | 2, 4, more |
| 安全性 | low, med, high |

となっている．変数の値に文字列が含まれているのでこれを数値にすると，

表 2　Car_evaluation の変換後のデータ

| 変数名 | 変数の値 |
|---|---|
| 総合評価 | 1, 2, 3, 4 |
| 買取価格 | 1, 2, 3, 4 |
| メンテナンス費 | 1, 2, 3, 4 |
| ドアの数 | 1, 2, 3, 4 |
| 乗車定員 | 1, 2, 3 |
| 安全性 | 1, 2, 3 |

と表 1 から表 2 のようにした．ドアの数や乗車定員は数値であるが見やすさのために表 2 のように置き換えた．
このデータの総合評価を目的変数，残りの変数を説明変数として学習を行う．このデータの特徴としては説明変数同士が完全に独立となっている点と総合評価のクラスごとのデータが不均衡になっている点があげられる．クラスごとのデータ数を表 3 に示す．

表 3　クラスごとのデータ数

| 総合評価の値 | 1 | 2 | 3 | 4 | 合計 |
|---|---|---|---|---|---|
| データ数 | 1210 | 384 | 69 | 65 | 1728 |

表 3 からわかるように，総合評価の値が 1 のデータ数が最も多いデータとなっている．このデータを 2 値分類問題にするため総合評価の値が 3，4 のデータを除いたデータで学習を行っていく．また，総合評価の値を見やすくするため 1 を 0，2 を 1 にする．つまり，表 4 のようにした合計 1594 個からなるデータで学習を行う．

表 4　学習に用いるデータ数

| 総合評価の値 | 0 | 1 | 合計 |
|---|---|---|---|
| データ数 | 1210 | 384 | 1594 |

### 6.2 提案手法の学習

包除積分ネットワークの学習は Python の Chainer(3)を用いて行う．学習の条件としては表 5 のようにする．

表 5　学習の条件

| | |
|---|---|
| **モデルの評価検証法：** | 5-分割交差検証 |
| **更新式：** | Adam |
| **誤差関数：** | Sigmoid_cross_entropy |

Car_evaluation データは表 4 のように不均衡なデータとなっているため Under Sampling と Over Sampling をそれぞれ行い精度の比較を行った．それぞれの Sampling 手法ではデータ比率が 1:1 となるようにする．つまり，Under_Sampling 手法ではデータ数が少数派クラスのデータ数×2 で 384×2=768，Over_Sampling 手法ではデータ数が

1 https://github.com/scikit-learn-contrib/imbalanced-learn/tree/master/imblearn

2 https://archive.ics.uci.edu/ml/datasets/car+evaluation

3 https://github.com/chainer/chainer

多数派クラスのデータ数×2 で 1210×2=2420 となる．表 6 はそれぞれの Sampling 手法での精度（正答率），適合率，再現率，F 値を閾値 0.5 とした時の値と ROC 曲線の面積である AUC を示している．

表 6　Sampling 手法による精度の比較

| | 精度 | 適合率 | 再現率 | F 値 | AUC |
|---|---|---|---|---|---|
| Sampling なし | 0.922 | 0.87 | 0.793 | 0.83 | 0.9821 |
| Cluster_Centroid | 0.969 | 0.953 | 0.987 | 0.97 | 0.9896 |
| Random_under | 0.949 | 0.929 | 0.974 | 0.95 | 0.985 |
| SMOTE | 0.951 | 0.93 | 0.976 | 0.952 | 0.9847 |

表 6 より Sampling しない場合より Under_Sampling や Over_Sampling をした場合のほうが精度がよく，その中でも Cluster_Centroid による Sampling 手法がどの指標でも最も高い結果となった．

### 6.3 提案モデルから得られるシャープレイ値,SHAP 値

6.2 で得られたモデルの内，最も精度のよくなった Cluster_Centroid でシャープレイ値を求めると表 7 のような結果が得られた．

表 7　提案手法から得られたシャープレイ値

| 買取価格 | メンテ費用 | ドアの数 | 乗車定員 | 安全性 |
|---|---|---|---|---|
| 2.83 | 3.06 | -3.95 | 9.66 | 10.26 |

表 7 の結果からドアの数は出力に貢献できておらず，最も出力に貢献できているのは安全性であることがわかる．

また,Python のライブラリとして提供されている shap(4)を用いることで SHAP の値を求めることができる．提案手法のモデルを shap の特別な重み付き線形回帰を使用して重要度を計算する KernelExplainer で重要度を求めると図 2 の結果が得られた．

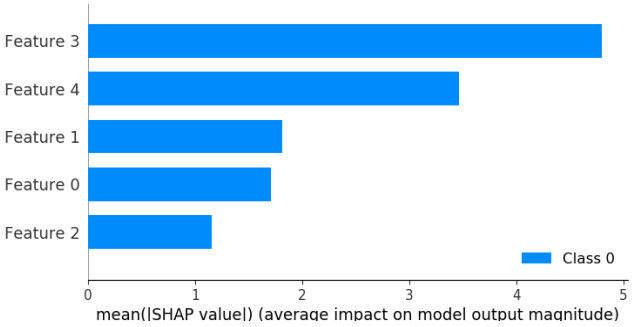


図 2　提案手法の学習モデルから得られた car_evaluation データの SHAP 値

図 2 の Feature0~4 は 0 から順位買取価格，メンテ費用，ドアの数，乗車定員，安全性である．図 2 から SHAP 値は乗車定員が最も大きく，ドアの数が最も小さい値となった．

### 6.4 XGBoost の学習

XGBoost は Python のライブラリとして提供されている xgboost(5)を使用して学習を行った．木の深さは最大 5 として学習を行わせ，閾値 0.5 として提案手法同様に精度を求めると表 8 の結果が得られた．

表 8　XGBoost により得られた精度

| | 精度 | 適合率 | 再現率 | F 値 | AUC |
|---|---|---|---|---|---|
| Sampling なし | 0.929 | 0.828 | 0.888 | 0.856 | 0.9835 |
| Cluster_Centroid | 0.977 | 0.955 | 1.000 | 0.977 | 0.9877 |
| Random_under | 0.948 | 0.920 | 0.982 | 0.950 | 0.9841 |
| SMOTE | 0.955 | 0.934 | 0.979 | 0.956 | 0.9918 |

表 8 より XGBoost では AUC を除いて Cluster_Centroid が最も良い精度となった．AUC だけを見ると SMOTE による Over_Sampling が最も高い数値となった．

### 6.5 学習した XGBoost モデルの重要度,SHAP 値

6.4 で AUC の値が最大となっている SMOTE による Over_sampling 時のモデルの重要度を gain で求めると表 9 のような結果が得られた．

表 9　XGBoost モデルの gain

| 買取価格 | メンテ費用 | ドアの数 | 乗車定員 | 安全性 |
|---|---|---|---|---|
| 6.31 | 6.81 | 1.67 | 25.59 | 26.32 |

表 9 よりドアの数の重要度が最小となっており安全性が最も重要となっている．また，xgboost は shap の TreeExplainer に引数として与えることができ重要度を図 3 のように求めることができる．

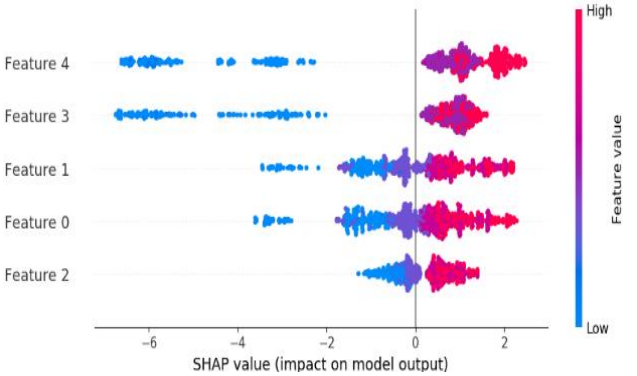


図 3 XGBoost の学習モデルから得られた car_evaluation の SHAP 値

図 3 から安全性，乗車定員は説明変数の大きさによって SHAP 値が正と負で分かれていることがわかる．

### 6.6 Car_evaluation データでの比較

表 8 と表 9 を比較するとどちらも各変数ごとの重要度の大小関係が同じとなっている．二つの表を割合で示すと，表 10 のようになった．

表 10　提案手法と XGBoost の重要度比較

| | シャープレイ値 | Gain |
|---|---|---|
| 買取価格 | 0.1295 | 0.0946 |

4 https://github.com/slundberg/shap

5 https://github.com/dmlc/xgboost

| | | |
|---|---|---|
| メンテ費用 | 0.14 | 0.1021 |
| ドアの数 | -0.1807 | 0.025 |
| 乗車定員 | 0.4419 | 0.3837 |
| 安全性 | 0.4694 | 0.3946 |

表 10 からどちらの重要度でもドアの数の重要度が低く買取価格とメンテ費用は重要度が同程度で乗車定員と安全性も重要度が同程度となっている．

### 6.7 Titanic データセットの準備

Car_evaluation データセットは各説明変数間が独立なデータセットであった．Titanic データセット(6)は kaggle より提供されているデータセットで，1912 年に起きたタイタニック号沈没の際の乗客のデータを取り扱っている．データの形は表 11 のようになっている．

表 11　Titanic データの内容

| | |
|---|---|
| Survived | 0(死亡)，1(生存) |
| Pclass | 1,2,3(旅客等級) |
| Name | 氏名(文字列) |
| Sex | male, female |
| Age | 年齢(整数値) |
| SibSp | 兄弟，配偶者の人数 |
| Parch | 親，子の人数 |
| Ticket | 数字を含む文字列 |
| Fare | 乗船料金 |
| Cabin | 数字を含む文字列 |
| Embarked | S, Q, C |

このデータは Suvived を目的変数，残りの変数を説明変数として予測を行う．表 11 のままでは文字列が含まれているので学習に使えないため，文字列を数値に直していく．まず氏名は，氏名の中に含まれている Mr,Ms などの敬称が含まれているので敬称ごとに数値化していく．表 12 は敬称を数値化した表である．

表 12　Name の数値化

| 敬称 | 敬称の数値化後の値 |
|---|---|
| Don, Rev, Capt | 1 |
| Mr | 2 |
| Major, Col, Dr | 3 |
| Mrs, Miss, Master | 4 |
| Mme, Ms, Lady, Sir, Mile, Aff | 5 |

表 12 では変数 Survived との相関係数が大きくなるように数値化している．性別は女性を 0，男性を 1 としている．SibSp と Parch はまとめて単独乗船という変数を作り単独で乗船していたら 1，誰かと同船している場合 0 としている．Cabin は頭文字で分け，頭文字が A,C,F ならば 1，B,D,E ならば 2 とし，それ以外を 0 とする．Ticket は頭文字が 1 ならば 1，2 ならば 2，3,S,C,A ならば 3 としそれ以外を 0 とする．Embarked は乗客の乗船した港の名前を省略して S,C,Q とあらわしており S:サザンプトン, C:シェルブール，Q:クイーンズタウンである．Suvived との相関が大きくなるように数値化して S を 1，C,Q を 0 とした．また，Cabin や Ticket は空欄の場合があるので空欄となっている場所はそれぞれの中央値とした．表 11 を数値化して書き換えたものを表 13 に示す．

表 13　数値化した Titanic データ

| | |
|---|---|
| 生死 | 0, 1 |
| 旅客等級 | 1, 2, 3, 4 |
| 敬称 | 1, 2, 3, 4, 5 |
| 性別 | 0, 1 |
| 単独乗船 | 0, 1 |
| 乗船料金 | 正の実数値 |
| 乗船港 | 0, 1 |
| 部屋番号 | 0, 1, 2 |
| チケット番号 | 0, 1, 2, 3 |

表 14　生死によるデータの割合

| 生死の数値 | 0 | 1 |
|---|---|---|
| データ数 | 549 | 342 |

このデータセットは Car_evaluation データと異なり各変数間は独立しておらず変数間で複雑な関係を持っており，学習データやラベルのない test データ内には空欄となっている部分が存在する．今回は空欄となっている部分は各変数ごとの中央値とした．また，表 14 のように死亡数のほうが生存数よりも多くなるような偏ったデータになっているため今回も Under_Sampling や Over_Sampling を行ってデータの割合が 1:1 となるようにして提案手法と XGBoost で学習を行い，それぞれシャープレイ値と gain を求める．

### 6.8 Titanic データによる提案手法の学習

Titanic データでも表 5 の条件で Under_sampling や Over_sampling を行ったデータでそれぞれ学習を行っていた結果が表 15 である．

表 15　提案手法による Titanic データに対する精度

| | 精度 | 適合率 | 再現率 | F 値 | AUC |
|---|---|---|---|---|---|
| Sampling なし | 0.811 | 0.851 | 0.631 | 0.718 | 0.844 |
| Cluster_Centroid | 0.756 | 0.838 | 0.640 | 0.722 | 0.821 |
| Random_under | 0.797 | 0.862 | 0.710 | 0.777 | 0.863 |
| SMOTE | 0.811 | 0.896 | 0.701 | 0.786 | 0.881 |

また，Titanic データには学習用のデータのほかに competition 用のデータ(test データ)があるので，そちらのデータで正答率を確認した．

表 16 提案手法による test データに対する精度

| | test データに対する精度 |
|---|---|

6 https://www.kaggle.com/c/titanic

| | |
|---|---|
| Sampling なし | 0.7847 |
| Cluster_Centroid | 0.7799 |
| Random_under | 0.7703 |
| SMOTE | 0.7799 |

表 15 と表 16 より AUC の値が高い場合でも test データによる正答率は高くない場合もあり今回 Sampling なしの正答率が最も大きい値となった．包除積分ネットワークでは通常出力層につながっているユニットの数は$2^8$となっておりすべての変数間の協力関係を計算可能になっている．次の実験では，この変数間の協力関係をすべてではなく 2 変数間までの協力関係を学習させた．例として説明変数の数が 3 つの場合は図 1 のようなネットワークであるが 2 変数までにすると，図 2 ようになる．

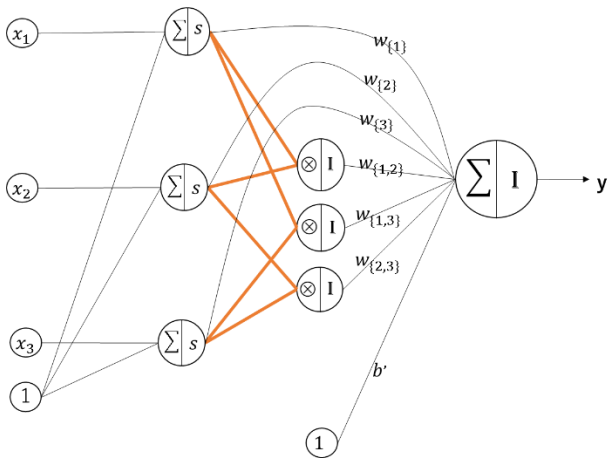


図 4 ユニットを減らした包除積分ネットワーク

Titanic データで出力層につながっているユニット数を 2 変数までの協力関係までに減らす，つまり出力層につながっているユニット数を 1 変数と 2 変数間とバイアスの数 ${}_8C_1 + {}_8C_2 + 1 = 37$とした時の学習結果を表 17 に示す．

表 17　2 変数間の協力関係までに制限した提案手法による Titanic データに対する精度

| | 精度 | 適合率 | 再現率 | F 値 | AUC |
|---|---|---|---|---|---|
| Sampling なし | 0.827 | 0.870 | 0.646 | 0.740 | 0.871 |
| Cluster_Centroid | 0.782 | 0.873 | 0.661 | 0.751 | 0.846 |
| Random_under | 0.798 | 0.856 | 0.720 | 0.780 | 0.873 |
| SMOTE | 0.827 | 0.892 | 0.743 | 0.810 | 0.897 |

同様に test データに対する精度を確認すると表 18 のような結果が得られた．

表 18　2 変数間の協力関係までに制限した提案手法による test データに対する精度

| | test データに対する精度 |
|---|---|
| Sampling なし | 0.7847 |
| Cluster_Centroid | 0.7943 |
| Random_under | 0.7656 |
| SMOTE | 0.7847 |

表 15～表 18 からすべての変数間の協力関係を学習させるより 2 変数間までに制限することで精度が向上していることがわかった．

### 6.9 Titanic データのシャープレイ値,SHAP 値

6.8 より最も精度のよいと考えられる 2 変数間の協力関係までに制限したうえで SMOTE の Over_sampling を行ったモデルでシャープレイ値を求めると表 19 の結果が得られた．

表 19　提案手法のシャープレイ値

| | シャープレイ値 | シャープレイ割合 |
|---|---|---|
| 旅客等級 | 2.18 | 0.2370 |
| 敬称 | 3.42 | 0.3717 |
| 性別 | 0.74 | 0.0804 |
| 単独乗船 | -0.15 | -0.0163 |
| 乗船料金 | -1.36 | -0.1478 |
| 乗船港 | 1.01 | 0.1098 |
| 部屋番号 | 1.48 | 0.1609 |
| チケット番号 | 1.88 | 0.2043 |

表 19 より敬称が最も生存に貢献していること，乗船料金は生存に最も貢献していないことが分かった．また同様に shap で SHAP 値を求めると図 5 の結果が得られた．

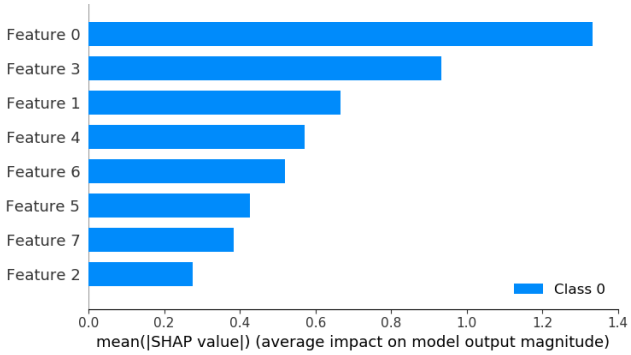


図 5 提案手法の学習モデルから得られた Titanic データの SHAP 値

Feature0~7 は 0 から旅客等級，敬称，性別，単独乗船，乗船料金，乗船港，部屋番号，チケット番号を表しており，これ以降の図の Feature0~7 は同様のものとする．図 5 から SHAP 値は旅客等級が最も大きく性別が最も小さい値となった．

### 6.10 Titanic データによる XGBoost の学習

XGBoost でも数値化した Titanic データを使って 6.4 でやったように学習を行っていく．Over_sampling や Under_sampling も同様に行い，それぞれの手法での精度を示しているのが表 17 である．

表 20　XGBoost による Titanic データに対する精度

| | 精度 | 適合率 | 再現率 | F 値 | AUC |
|---|---|---|---|---|---|
| Sampling なし | 0.835 | 0.807 | 0.749 | 0.776 | 0.8814 |

| | | | | | |
|---|---|---|---|---|---|
| Cluster_Centroid | 0.788 | 0.788 | 0.786 | 0.786 | 0.8698 |
| Random_under | 0.811 | 0.818 | 0.799 | 0.808 | 0.8611 |
| SMOTE | 0.843 | 0.852 | 0.833 | 0.841 | 0.9160 |

test データによる正答率は表 18 のようになった.

表 21　XGBoost による test データに対する精度

| | test データに対する精度 |
|---|---|
| Sampling なし | 0.7751 |
| Cluster_Centroid | 0.7225 |
| Random_under | 0.7656 |
| SMOTE | 0.7799 |

表 17 と表 18 より SMOTE による Over_sampling が今回最も良い精度を出していることがわかった.

### 6.11 Titanic データの重要度,SHAP 値

6.10 より XGBoost では SMOTE による Over_sampling の精度が高いため SMOTE を行った時の XGBoost モデルの重要度を求めると表 19 のような結果が得られた.

表 22　Titanic データの gain

| | gain 値 | gain 割合 |
|---|---|---|
| 旅客等級 | 7.979 | 0.1963 |
| 敬称 | 24.065 | 0.5919 |
| 性別 | 1.703 | 0.0419 |
| 単独乗船 | 0.877 | 0.0216 |
| 乗船料金 | 1.234 | 0.0304 |
| 乗船港 | 1.356 | 0.0334 |
| 部屋番号 | 1.952 | 0.048 |
| チケット番号 | 1.492 | 0.0367 |

表 19 より敬称が生存において全体の 6 割程も重要だとされていることと，単独乗船か否かは生存には重要でないことがわかる．また shap で重要度を求めると図 6 の結果が得られた.


図 6XGBoost による Titanic データの SHAP

図 6 から旅客等級と単独乗船と，乗船港は変数の大きさにより SHAP 値が正と負におおよそ分かれていることがわかった．ここでシャープレイ値の割合と gain の割合を横に並べてみると表 23 のようになる.

表 23　シャープレイ値の割合と gain の割合の比較

| | シャープレイ割合 | gain 割合 |
|---|---|---|
| 旅客等級 | 0.2370 | 0.1963 |
| 敬称 | 0.3717 | 0.5919 |
| 性別 | 0.0804 | 0.0419 |
| 単独乗船 | -0.0163 | 0.0216 |
| 乗船料金 | -0.1478 | 0.0304 |
| 乗船港 | 0.1098 | 0.0334 |
| 部屋番号 | 0.1609 | 0.048 |
| チケット番号 | 0.2043 | 0.0367 |

## 7.　考察

実験結果よりデータによって効果的な Sampling 手法は異なること, Car_evaluation のようなデータでは提案手法のシャープレイ値と XGBoost の gain の値はどちらも高い順に安全性，乗車定員，メンテナンス費用，買取価格，ドアの数の順で総合評価に貢献, 重要であることが分かったが, Titanic データのシャープレイ値と gain を比較すると敬称, 旅客等級までは同じでそれ以降は異なる結果となった．この結果より，Car_evaluation では安全性，乗車定員，メンテナンス費用，買取価格，ドアの数の順で総合評価に重要であること，Titanic データでは少なくとも敬称，旅客等級の順で重要であることが分かった．また，提案手法のシャープレイ値と SHAP 値を比較すると，car_evaluation では変数間の重要度の大小関係は一致しており，Titanic データでは変数間の大小関係はあまり一致していなかった.これは shap の kernelExplainer で得られる重要度は提案モデルの近似モデルを作成する際にモデルの特徴が抜け落ちたためと考えられる.

## 参考文献

[1] 本田あおい, Theory of inclusion-exclusion integral,. Information Sciences 376(134-147), 2017.

[2]Chen, Tianqi. and Guestrin, Carlos.. XGBoost: A Scalable Tree Boosting System. KDD'16:Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, p.785-794.

[3] Lundberg, Scott M and Lee, Su-In, A Unified Approach to Interpreting Model Predictions, Advances in Neural Information Processing Systems 30, 2017.

[4]Nitesh, V. Chawla. Kevin, W. Bowyer. Lawrence, O. Hall. W. Philip, Kegelmeyer, SMOTE:Synthetic minority over-sampling technique.Journal of Artificial Intelligence Research,2004, Vol. 16, p20-29.